

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
REQUEST FOR FILING APPLICATION UNDER RULE 1.53(b)**

Pursuant to 37 CFR 1.53(b), please file a ☒ continuation/☐ divisional  
of the pending prior PATENT APPLICATION of:

Inventor: NATHAN et al.

Serial No. 09/144,440

Filed: September 1, 1998

For: **SYSTEM FOR REMOTE LOADING OF OBJECTS OR FILES IN ORDER TO UPDATE  
SOFTWARE**

Assistant Commissioner for Patent

Washington, DC 20231

Sir:

Atty Dkt.: 871-92

C# M#

Date: October 13, 2000

Group: 2762

Examiner: K. Booker

JC685 U.S. PTO



10/13/00

JC685 U.S. PTO  
09/689726



This request for filing under Rule 53(b) is made by the following named inventor(s) (using the above-identified title):

Inventor(s): NATHAN et al.

- ☒ Attached is a true copy of the prior application as originally filed including the certified English translation and 19 pages of the French version specification, claims, Oath/Declaration and drawings (if any) and abstract (if any). No amendments (if any) referenced in the Oath or Declaration filed to complete the prior application introduced new matter.
- ☒ Priority is hereby claimed under 35 USC 119 based on the following foreign applications, the entire content of which is hereby incorporated by reference in this application:

**Application Number**

**Country**

**Day/Month/Year/Filed**

98 08296

France

21 July 1998

☐ certified copy(ies) of foreign application(s) attached or

☐ already filed on \_\_\_\_\_ in prior appln. no. \_\_\_\_\_ filed \_\_\_\_\_

☒ already filed in 09/144,440 filed September 1, 1998

☒ Please amend the specification by inserting before the first line: -- This application claims the benefit of U.S.

Provisional Application No. \_\_\_\_\_, filed \_\_\_\_\_, the entire content of which is hereby incorporated by reference in this application.--

☒ The prior application is assigned to Touchtunes Music Corporation.

☒ Power of Attorney has been granted to Joseph S. Presta et al, Reg. No. 35,329 of Nixon & Vanderhye P.C., 1100 N. Glebe Rd., 8<sup>th</sup> Flr, Arlington, VA 22201.

☒ Address all future communications to: Nixon & Vanderhye P.C., 1100 N. Glebe Rd., 8<sup>th</sup> Floor, Arlington, VA 22201.

☒ Please amend the specification by inserting before the first line --This is a continuation of application Serial No. 09/144,440, filed September 1, 1998, now pending, the entire content of which is hereby incorporated by reference in this application.--

☒ The Examiner's attention is directed to the prior art cited in the parent application by applicant and/or Examiner for the reasons stated therein.

☒ Please enter the attached and/or below preliminary amendment **prior** to calculation of filing fee:

☒ **PTO-1449 Forms**

☒ The entire disclosure of the prior application above-referenced is considered as being part of the disclosure of this new application and is hereby incorporated by reference therein.

**FILING FEE IS BASED ON CLAIMS AS FILED LESS ANY HEREWITH CANCELED**

Basic Filing Fee				\$	710.00
Total effective claims	3	- 20 (at least 20) =	0	x \$ 18.00	\$ 0.00
Independent claims	1	- 3 (at least 3) =	0	x \$ 80.00	\$ 0.00
If any proper multiple dependent claims now added for first time, add \$270.00 (ignore improper)				\$	0.00
				<b>SUBTOTAL</b>	\$ 710.00
If "small entity," then enter half (1/2) of subtotal and subtract				\$(	0.00)
				<b>SECOND SUBTOTAL</b>	\$ 710.00
Assignment Recording Fee (\$40.00)				\$	0.00
				<b>TOTAL FEE ENCLOSED</b>	\$ 710.00

Any future submission requiring an extension of time is hereby stated to include a petition for such time extension. The Commissioner is hereby authorized to charge any deficiency in the fee(s) filed, or asserted to be filed, or which should have been filed herewith (or with any paper hereafter filed in this application by this firm) to our **Account No. 14-1140**. A duplicate copy of this sheet is attached.

1100 North Glebe Road, 8<sup>th</sup> Floor  
Arlington, Virginia 22201-4714  
Telephone: (703) 816-4000  
Facsimile: (703) 816-4100  
JSP:mg

**NIXON & VANDERHYE P.C.**

By Atty: Joseph S. Presta, Reg. No. 35,329

Signature: \_\_\_\_\_

VERIFICATION OF TRANSLATION

I, Tom Appich, a translator with Chillson Translating Service, 3326 Brantly Road, Glenwood, Maryland 21738, hereby declare as follows:

That I am familiar with the French and English languages;  
That I am capable of translating from French to English;  
That the translation attached hereto is a true and accurate translation of the French Application titled, "System For Remote Loading of Objects Or Files in Order To Update Software;"

That all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true;

And further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any registration resulting therefrom.

By Thomas W. Appich

Executed this 1 day of Sept. 1998.

Witness John W. Miller

JCE25 U.S. PRO  
09/689726  
10/13/00

SCANNED - 12

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re Patent Application of

NATHAN et al.

Atty. Ref.: 871-92

Serial No. Not Yet Assigned

Group: 2762

Filed: October 13, 2000

Examiner: K. Booker

For: SYSTEM FOR REMOTE LOADING OF  
OBJECTS OR FILES IN ORDER TO UPDATE  
SOFTWARE

\* \* \* \* \*

October 13, 2000

Assistant Commissioner for Patents  
Washington, DC 20231

Sir:

**PRELIMINARY AMENDMENT**

In order to place the above-identified application in better condition for  
examination, please amend the application as follows:

**IN THE ABSTRACT**

Please substitute the Abstract of the Disclosure submitted herewith on a separate  
sheet for the Abstract currently on file.

**IN THE SPECIFICATION**

Please amend the specification as follows:

Page 1, between lines 2 and 3, insert the following heading:

--BACKGROUND OF THE INVENTION--;

line 13, delete “said” and insert --the--.

Page 2, line 7, delete “a last”;

between lines 14 and 15, insert the following heading:

--SUMMARY OF THE INVENTION--;

line 20, delete “be”.

Page 4, line 11, delete “said” and insert --the--;

line 15, delete “said” and insert --the--;

line 21, delete “said” and insert --the--.

Page 5, line 4, delete “that”;

line 9, before “the” (first occurrence) insert --and--.

Page 6, between lines 22 and 23, insert the following heading:

--BRIEF DESCRIPTION OF THE DRAWINGS--.

Page 7, line 2, after “invention;” insert --and--;

between lines 4 and 5, insert the following heading:

--DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS--;

line 8, delete “Microprocessor” and insert --Referring to Figure 1, a  
microprocessor--;

line 24, delete “means” (both occurrences).

Page 8, line 3, delete “Bulk” and insert --A bulk--; delete “means”; delete “use”  
and insert --uses--;

line 4, delete “are” and insert --is--;

line 5, delete “means”; delete “are” and insert --is--;

line 6, delete “These means serve” and insert --This storage serves--;

line 19, delete “initialization means” and insert --display--; after “circuit”

insert --6--.

Page 9, line 2, delete “makes it possible, after having displayed” and insert

--is able to display--.

Page 10, line 25, delete “means” (both occurrences); after “reproduction” insert --

structure--.

Page 11, line 8, delete “said” and insert --the--.

Page 12, line 19, after “tasks,” insert --using the--.

## **IN THE CLAIMS**

Please cancel claims 1-15 and add new claims 16-18 as follows:

--16. A system for remote loading of objects or files in order to update software, comprising operating system architecture that includes a breakdown of various tasks into software modules that are interconnected by dynamic links or are made up of executable subroutines that have primary dependence links to other parts of the operating system, wherein each of the software modules is composed of object files or library files that are represented by dynamic link libraries that are organized between them according to a number of levels of dependence that are described in their respective attributes.--

--17. A system for remote loading of objects or files in order to update software according to claim 16, wherein the attributes of an object or a library indicate a version number and dependencies of the object with respect to other objects.--

--18. A system for remote loading of objects or files in order to update software according to claim 17, wherein the attributes indicate the level attributed to the software modules.--

### **REMARKS**

This Preliminary Amendment is being filed concurrently with the instant continuation application. Favorable examination is respectfully requested.

In the parent application, the Examiner asserted that Schmidt anticipates claims 17-19. For at least the following reasons, Applicant respectfully submits that the pending claims are patentably over the prior art of record.

Schmidt teaches a software version management system, which allows collecting and recompiling, automatically, an updated version of component software objects of a software program (see abstract). Applicant respectfully points out that the purpose and operation of the instant invention is significantly different from that of Schmidt. Specifically, the purpose of the present invention (see page 2 of specification) is to avoid rebooting the terminal when an updated component software is downloaded. According to claim 17, updating of the software is made with software modules or executable subroutine.

Further, the system of Schmidt concerns the development a program. According to this reference, the program components are stored in different locations in a network. All program components are accessible by a plurality of network users. This particular structure allows a plurality of users to access and modify each component. The system of

Schmidt insures that users are using always an updated version of the program and manages the modification introduced by a user.

On the contrary, the present invention is directed to a system for remote loading of objects or files in order to update software. This means that the module constituting the software of the terminal is not modified but is simply replaced by an updated one. In other words, there is no remote modification of modules as disclosed in Schmidt.

Further, all modules are stored on the terminal, which means the entire operating system is stored on the terminal, and not in different location stations of a network.

Schmidt teaches that the system uses modules, which are distinct components for managing the version and dependencies of the component (abstract). On the contrary, according to claim 17 of the present invention, the management of the dependency of the module between each other is integrated in each module itself by attributes. Thus, when a new module is downloaded, the checking of its dependency is made by the reading of the attributes comprising all dependence of this module.

Thus, Schmidt does not teach or suggest each and every element of the claimed invention, as required for anticipation under section 102. For example, Schmidt does not teach that modules used to update the software program are software modules or executable subroutines when they are downloaded on the terminal. Further, Schmidt does not teach that all modules are stored in a unique location. Moreover, none of the other references of record make up for the deficiencies in Schmidt.

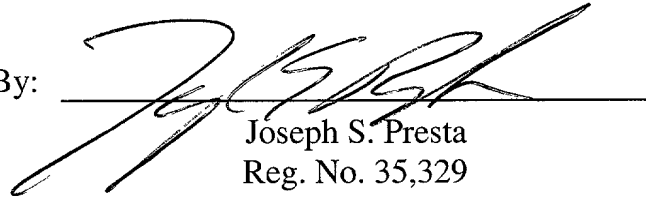
In view of the above, Applicant respectfully submits that all of the pending claims are in condition for allowance. Thus, allowance of the pending claims are earnestly solicited.

Should the Examiner have any questions, or deem that any further issues need to be addressed prior to allowance, the Examiner is invited to call the undersigned attorney at the phone number below.

Respectfully submitted,

**NIXON & VANDERHYE P.C.**

By: \_\_\_\_\_



Joseph S. Presta  
Reg. No. 35,329

JSP:mg  
1100 North Glebe Road, 8<sup>th</sup> Floor  
Arlington, VA 22201-4714  
Telephone: (703) 816-4000  
Facsimile: (703) 816-4100



## **ABSTRACT OF THE DISCLOSURE**

A system for remote loading of objects or files in order to update software includes operating system architecture that calls for different tasks to be broken down into software modules that are connected to one another by dynamic links or are composed of executable subroutines that have main dependence links to other parts of the operating system. Each of the modules is composed of object files or libraries that are represented by dynamic link libraries among themselves according to a number of dependence levels that are described in their respective attributes.

465668

**System For Remote Loading of Objects Or Files in Order**  
**To Update Software**

This invention pertains to a system for remote loading of objects or files in order to update software, particularly for audiovisual reproduction systems that are triggered by the payment of fees, such as jukeboxes or other devices.

In the prior art, devices for remote loading of an operating system through a network are known such as, for example, from British Patent No. 2 231 180. The teaching of this patent application calls for loading a first computer, which is connected to a second computer via a network, by loading a subset of the operating system into the memory of the first computer, whereby said subset contains the commands that make it possible to copy files, create a directory, and format a disk, as well as allowing a connection to operate through the network so that this subset can then be used to transfer all of the operating system files from the second computer to the disk of the first computer.

In this type of remote loading, the purpose is to load the entire operating system and all of the operating system files. This thus limits remote loading either to tying up, for relatively long periods of time, telecommunications systems that are to carry out remote loading in the case of the operating system, or causes the relevant files to have to be updated frequently.

From U.S. Patent No. 4,958,278 a system is also known for remote loading to a terminal that is not equipped with a disk player.

From French Patent No. 2 682 786 another process is known for remote loading to a terminal that is not equipped with a hard disk.

Finally, a last European Patent No. 0 358 992 teaches a method for remote loading, via a network, of an operating system or of executable files to a computer that does not include a boot device or other devices that hold the executable program. A first minimum boot program is transferred initially, and then this minimum boot program executes itself, establishes a logical link to a disk of the server, and allows the querying computer to treat the server disk as a local boot device.

The object of the invention is to avoid the necessity, on the one hand, of rebooting the terminal to which downloading is done and, on the other, to make it possible to transfer operating files or parts of an executable program without having to reinitialize the machine, and doing so while making sure that the operation of the system is not be degraded by the remotely loaded version.

This object is achieved by virtue of the fact that the architecture of the operating system provides for breaking the different tasks down into software modules that are interconnected by means of dynamic links or are composed of executable subroutines that have main dependence links to other

parts of the operating system, whereby each of the modules is composed of object files or libraries that are represented by dynamic link libraries, which are organized among themselves according to a number of dependence levels, as described in their respective attributes.

According to another feature, the attributes of an object or a library indicate the version number and the dependencies of the object with respect to other objects.

According to another feature, the attributes indicate the levels attributed to the modules.

According to another feature, the different tasks include a main task that includes a module (JUKECORE), which is designed to load the dynamic link libraries (DLLs), to initialize the nucleus of the program, to initialize the graphics management module (GFX), to initialize the library loading module (WDLL), to load the Telecom module of the telecommunications tasks (TELECOM.DLL), and to launch the screen interpreter program as a main task.

According to another feature, the program is subdivided into a certain number of modules that define a task that is specific to the terminal.

According to another feature, this specific task is that which corresponds to a jukebox and includes a specific main task, a certain number of modules that define the "Windows" functions of the display, which are the following:

- a module for running the mouse signals or the touch screen;

- a module for running the messages that are exchanged among the objects;

- a "FIL.DJL" module for managing the files on disk;

- a "FIL.DJL" module for reading and writing files to and from disk;

- a "FILIO.DJL" module for monitoring all of the events that are generated by the hardware, such as the touch screen, the sound card, and the money hardware processing interface.

According to another feature, the main task of the jukebox application includes a "SILOAD.DLL" module, which contains the library of the loader program, whereby said library is intended to verify the versions of the dynamic link libraries that are requested, to load them, and to call the Telecom task modules in order to transfer files.

According to another feature, said SILOAD module includes the list in a file (DLL.DEFAULT) of the minimum versions that are required for operation, as well as the list of all of the functions that are represented either by the libraries (DJL) (DATA JUKEBOX LIBRARY) or by the object files (DJO DATA JUKEBOX OBJECT).

According to another feature, said object or library contains the list of all of the functions that the library or object needs, as well as the version numbers and dependencies.

According to another feature, WDLL ensures the management of all of the new modules and verifies that the remotely loaded modules do not have any missing dependencies and that they have

been loaded with the necessary versions.

According to another feature, SILOAD manages the loading of the modules that are specific to the task of the terminal, i.e., all of the "DJL" modules already listed above, as well as that the jukebox library modules constituted by WOBJECT manage the object, the mixer, and the purchases; the "WCURSOR" module manages the movements of the cursor; the DBMAPI module manages the database; the "WFORMATS" module manages all of the font types; the "PARSER" module analyzes and generates the screens from the script and verifies the grammar with the aid of the "GRAMMAR.DJL" module and the lexical module "LEXY.DJL."

According to another feature, the library loading module SILOAD includes a "WINDEF" module that contains the list of the files that have to be included in order to manage the windows of a Windows display that is supplied on the monitor of the jukebox-type terminal.

According to another feature, this list of objects consists of:

- an "OBJET WPSCREEN.DJO" module, which makes it possible to define the main page on the monitor;

- a "WSCREEN.DJO" object module, which makes it possible to determine in this main page the number of screens that are available and thus to allow multiple windows or screens to be displayed;

- a "WIMAGE.DJO" module, which makes it possible to determine and define on the screen the image that it will use;

- a "WANIM.DJO" module, which makes it possible to define the animation when the image is animated;
- a "WBUTTON.DJO" module, which makes it possible to define and manage the buttons that are used on the screen of the main page;
- a "WLABEL.DJO" module, which makes it possible to create the labels that make it possible to write on top of an object; and
- a "WSCROLLER.DJO" module, which makes it possible to design the scroll display zones, between two points for example, horizontal, diagonal, vertical.

According to another feature, all of these object modules, which are managed by the main task, use a "JHANDLER" library, which makes it possible to define the fixed uses of the screens and thus to determine which are the interfaces that ensure the link to the different objects that are defined by the preceding modules.

According to another feature, the SILOAD task launches or loads the "XCP" module, makes it possible to manage payment tasks such as those handled by ticket receiving systems or coin or card payment units, and also makes it possible to save the basic information in the IBUTTON.

Other features and advantages of this invention will be made clearer by reading the description given below, with reference to the attached drawings, where:

Figure 1 shows a schematic of the electrical diagrams of the hardware that constitutes the invention;

Figure 2 shows a flow chart of the relationships between the library modules and the object modules.

Preferably, but without being limited hereto, the audiovisual reproduction system uses the hardware elements listed and referenced below.

Microprocessor central unit 1 is a high-performance PC compatible system, whereby during implementation a Pentium-type system was chosen that has the following memory resources and specifications:

- compatibility with the local VESA bus;
- processor cache: 256 KB minimum;
- RAM memory: 32 MB;
- high-performance serial ports;
- SVGA microprocessor graphics adapter;
- SCSI/2 type bus controller;
- self-powered static RAM memory.

Any other central processing unit that has the equivalent or better performance can be used in the invention.

This central processing unit controls and manages a sound control circuit (5), a telecommunications control circuit (4), an input control circuit (3), a bulk storage control circuit (2), and a display means control circuit (6). The display means consists primarily of a flat-screen non-interlaced SVGA high-resolution and low-radiation video monitor (62), i.e., this



is the monitor that is used to reproduce images (for example, the album covers of musical selections), graphics, or video clips.

Bulk storage means (21), which use hard disks of the high-speed and high-capacity SCSI type, are associated with the storage means that are already present in the microprocessor device. These means serve to store digitized and compressed audiovisual information.

A high-speed telecommunications modem adapter (41), which is built in in order to enable the link to an audiovisual information distribution network that is controlled by a central server.

In order to reproduce the sound information of the musical selections, the system includes loudspeakers (54) that receive the signal from a tuner-amplifier (53), which is connected to electronic circuit (5) that incorporates two memory buffers (56, 57) of the music synthesizer type that are supplied to support a large number of input sources while providing CD (compact disc) type output quality.

Likewise, the initialization means control circuit also has two buffer memories (66, 67) for the purpose explained above.

A 240-watt thermally regulated and ventilated power supply feeds power to the system. This power supply is protected against overcurrents and overdriving.

By means of its input controller circuit (3), the audiovisual reproduction system manages a touch screen (33), which includes a glass coating panel that uses the "advanced

surface-wave technology", as well as an AT-type bus controller. This touch screen makes it possible, after having displayed on video monitor (62) or the screen of a television set (61) various selection data that are used by the customers, as well as management command and control data that are used by the manager or owner of the system. This touch screen is also used for maintenance purposes in combination with an external keyboard (34) that can be connected to the system which, for this purpose, has a keyboard connector that is controlled by a key lock (32) via interface circuit (3).

Input circuit (3) also interfaces the system with a remote control assembly (31) that consists of, for example, a radio frequency RF remote control.

A fee payment device (35) is also connected to input interface circuit (3). It is also possible to use any other device that makes it possible to receive any kind of payment using coins, tickets, tokens, magnetic or microchip cards, or any combination of these means of payment.

To allow the system to be installed, it is equipped with a chassis or is built with customizable external fittings.

In addition to these elements, a wireless microphone (55) is connected to sound controller (5); this makes it possible to convert the sound controller into a powerful public-address and information system or optionally into a karaoke machine. Likewise, the system can also use a wireless loudspeaker system.

Remote control assembly (31) allows the manager from, e.g., behind the bar, to access and control various commands such as:

- the microphone on-off control;
- the loudspeaker muting control;
- the control for canceling the musical selection that is currently being played.

Two buffers (56, 57) are associated with sound controller circuit (5) in order to make it possible to store, each alternately, a data item corresponding to at least approximately a quarter of a second of sound. Likewise, two buffers (66, 67) are associated with video controller circuit (6), whereby each buffer is able, either by itself or alternatively, to store at least approximately one-tenth of a second of an image. Finally, a respective buffer (46, 36, 26) is associated with each of communications controller circuit (4), input interface circuit (3), and storage circuit (2).

The system operating software was developed around a library of tools and services that were very largely oriented toward the audiovisual domain in a multimedia universe. This library advantageously includes a high-performance multitask operating system that effectively allows the simultaneous execution of multiple fragments of code. This operating software also allows the concurrent execution, in an orderly and completely conflict-free way, of operations that are carried out on the display means and the sound reproduction means, as well as the managing of the telecommunications links through the distribution

network. Moreover, this software is highly flexible.

The operating system is divided into modules, which include a first boot module (7), which in turn is subdivided into a first main program module (70) "JUK.EXE", which checks the memory and verifies whether the minimum number of objects is available to ensure the operation of the jukebox; a second module (71), which is dynamically linked and is dependent thereon, is represented by the "JUKECORE.DLL" module. The function of said second module (71) is to hold the libraries in C and to ensure the execution of the main task.

The architecture of the operating system calls for the different tasks to be broken down into software modules that are connected to one another by dynamic links or consist of executable subroutines that have main dependence links to other parts of the operating system. Each of the modules is composed of object files or dynamic link library files that are organized according to a number of dependence levels described in the attributes. The attributes of an object or a library indicate the version number and the dependencies of the object or library file with respect to other object files, as described below for the PARSER module. Each attribute indicates the level attribute to the module. Thus, the JUK.EXE module (70) is of a higher level than the modules JUKECORE (71), TLS (72), CRDE (73), GFX (74), WDLL (75), JEEP (9) and TELECOM (10), but TELECOM module (10) is dependent on JEEP module (9) (see link 910), and is thus lower in level than JEEP (9).

Likewise, JEEP (9) (see link 759) is of a lower level than WDLL module (75), because it is dependent on it and TLS (725), of a higher level than WDLL (75). However, TLS and GFX may be on the same level. The main task includes a module (JUKECORE) whose purpose is to initialize or load module (73), the nucleus of program "CRDE.DLL", to initialize or load module (74) for managing graphics (GFX), to initialize or load module (75) for loading libraries (WDLL.DLL), to load Telecom module (10) for the task of telecommunications (DLL), to load TLS.DLL module (72), which contains all the uses required for the jukebox, for telecommunications, time, decryption, etc.,..., to initialize or load the library of JEEP (Juke Execution Exchange Protocol) programs, which handle the tasks of an integrity server, load request and dialogue with the server, and to launch program (80, SILOAD.DLL) as a main task. The main task of the jukebox application includes a module (SILOAD.DLL) that contains the library of the loader program whose purpose is to verify the dynamic link library versions required in (WDLL), to load them or to call the tasks, Telecom module in order to transfer files. The WDLL.DLL module includes the list in a file (DLL.DEFAULT) of the versions that are required for operation, as well as the list of all of the functions that are represented by libraries (LIBRARY) (DLL, DJL) or by object files (DJO). Each object or library contains the list of all of the functions that the library or object needs, as well as the version members and dependencies. The WDLL module manages all of the new modules,

checks to verify the interdependencies, and verifies that the remotely loaded modules have no dependence and have been loaded with the required versions. Application part (8) that is inherent in a jukebox includes a certain number of modules that are loaded and launched by SILOAD and that define the "Windows" functions of the display, which include the following:

- a module (81) for running the mouse or touch-screen signals;
- a module (82) for running the messages that are exchanged between the objects and the various other modules;
- a FIL.DLL module (83) for managing the files on disk;
- a read-write module (84) FILO.DJL for files on disk;
- a JSTRUCT.DJL module (85) for monitoring all of the events produced by the hardware, such as the touch screen, the sound card, and the interface for processing fee hardware.

SILOAD manages the loading of the modules that are specific to the task of the terminal, i.e., all of the DJL modules already listed above, as well as jukebox library modules (87), consisting of WOBJECT (870), which manages the objects such as the mixer and purchases; WCURSOR module (871), which manages the movements of the cursor; DBMAPI module (872), which manages the database; WFONTS module (873), which manages all of the types of fonts; PARSER (syntactic analysis program) module (874), which analyzes and generates the screens from the script and verifies grammar with the aid of module "GRAMMAR.DJL" (876) and module "LEXY.DJL" (875), which is the lexical module that assigns the functions of

the words within the language. PARSEr module (874) contains in its file header the following information:

```
-char*parser _ version _ info = "DLL_INFO.DJL,"  
                                "DLL-NAME PARSEr.DJL"  
                                "VERSION 1;"  
                                "CREATOR KENDALF;"  
                                "REQUIRES lexyy.djl;4"  
                                "REQUIRES grammar-djl;5";
```

All of the modules and all of the libraries (DJO, DLL, DJL) contain information similar to that of the PARSEr module, and this information determines the needs in terms of versions and dependence.

Thus, the PARSEr module needs LEXY module version 4 and GRAMMAR module version 5 in order to allow it to be executed by the system. The double arrows in Figure 2, which connect the various modules to one another, indicate the order in which the different files are loaded. Thus, as indicated previously, it will be necessary to start by loading JUKE.EXE, then loading JUKECORE.DLL, and being able, from JUKECORE.DLL, to load GFX.DLL, TLS.DLL, WDLL.DLL, JEEP.DLL, TELECOM.DLL, CRDE.DLL, and SILOAD.DLL.

The single arrows represent the dependencies between files.

Thus, arrow (91) indicates that the DJL and, in particular, DBMAPI modules are dependent on CRDE.DLL. Arrow (93) indicates that the DJO files are dependent on the WOBJECT.DJL module. The WOBJECT.DJL module is itself dependent on the FILIO.DJL module.

Arrow (92a) indicates that DBMAPI.DJL is dependent on JSTRUCT.DJL, and arrow (92b) indicates that DBMAPI.DJL is dependent on WMESSAGE.DJL. Arrow (98) indicates that JSTRUCT.DJL is dependent on the WMESSAGE.DJL file. WMESSAGE is dependent on the MOUSE.DJL file and, since FILIO.DJL is dependent on the FIL.DJL file, the file XCP.DJL is dependent, as indicated by arrow (856), on JSTRUCT.DJL and, as indicated by arrow (826), on WMESSAGE.DJL. The JHANDLER file is dependent, as indicated by arrow (97), on WMESSAGE.DJL and, as shown by arrow (96), on JSTRUCT.DJL. The SILOOP.DJL file is dependent, as indicated by arrow (95), on JSTRUCT.DJL and, as indicated by arrow (94), on WMESSAGE.DJL. The TELECOM.DLL file is dependent, as indicated by arrow (910), on JEEP.DLL, which in turn is dependent, as shown by arrow (959), on WDLL.DLL. The WDLL.DLL file is dependent, as indicated by arrow (725), on TLS.DLL. Likewise, arrow (89c) shows that GRAMMAR.DJL is dependent on LEXY.DLL and, as indicated by arrow (99b), that LEXY.DJL is dependent on PARSER.DJL. Thus, as was seen previously, PARSER requires LEXY and GRAMMAR to execute itself, and version 1 of PARSER calls version 4 of LEXY.DJL and version 5 of GRAMMAR.DJL. Likewise, WOBJECT.DJL is dependent, as indicated by arrow (99a), on PARSER.DJL. Thus, all of the .DJO, .DLL, and .DJL modules and libraries contain information similar to that of the PARSER module, which determines the version requirements of the various modules on which a given module is dependent. This information also indicates the dependencies of the modules with respect to the



other modules, as indicated by the arrows in Figure 2.

Library loading module SILOAD also loads or launches a module SILOOP.DLL (90), with an event wait tape. A set of modules (88) contains a list of the files that have to be included there to manage the windows of a Windows display that is provided on the monitor of the jukebox-type terminal.

This list of objects consist of:

- an object file (883) "WPSCREEN.DJO", which makes it possible to define the main page on the monitor;

- an object file (881) "WSCREEN", which makes it possible to determine on this main page the number of screens available and thus to make it possible to display multiple windows or screens;

- an object file (880) "WIMAGE.DJO", which makes it possible to determine and define on the screen the image that the screen will use;

- an object file (882) "WANIM.DJO", which makes it possible to define the animation when the image is animated;

- an object file (885) "WBUTTON.DJO", which makes it possible to define and manage the buttons that will be used on the screen of the main page, such as the actuation buttons used in the graphical interface that is defined in patent application PCT WO 96/12258;

- an object file (884) "WLABEL.DJO", which makes it possible to create labels that make it possible to write on top of an object; and

- an object file (886) "WSCROLLER.DJO", which makes it possible to define the vertical-scroll display zones.

A "JHANDLER" library makes it possible to define the fixed uses of the screens and thus to determine which are the interfaces that ensure the links to the different objects that are defined by the previous modules.

Library module "XCP" (86) makes it possible to manage the payment tasks such as those of ticket receiving systems or coin or card payment units and also makes it possible to back up basic information in the IBUTTON, which is an integrated circuit for storing secret codes for the user.

Thus, when a new file is sent by remote loading to the system, the file contains information on its level, which depends on the type of file. The files of the graphical images, for example WIMAGE.DJO, have the highest levels, and the hardware management files, for example XCP.DJL, have the lowest levels. The JEEP module verifies the dependence logic by starting with the lowest-level files and moving up toward the higher files, all the while ensuring that the necessary dependencies between the files or libraries are respected. In this way, a modification in WOBJECT.DJL will cause JEEP to verify that the version information required by WOBJECT.DJL for the DJO files that are dependent and are required for its execution corresponds to the minimum versions required by the information recorded in the WOBJECT.DJO file. Thus, if WOBJECT.DJL requires a certain version of WPSCREEN.DJO, it will verify that this version is at

least present and that, if there is only a version of an inadequate level, it will report a problem. Then JEEP will go up the dependence links toward FILIO.DJL and FIL.DJL.

The hard disk of the jukebox is organized in such a way as to comprise a directory C:\NEWJUKE, which contains the new jukebox files when new modules are remotely loaded. Another file, C:\OLDJUKE, contains a backup of the stable versions of the files and modules that are required for the operation of the jukebox. The JEEP (JUKE EXECUTION EXCHANGE PROTOCOL) module contains an automatic file manager that keeps track of the modules and files that are updated by backing up the old files for a certain period of time and by moving their files into the NEWJUKE directory. This task also copies the files on the tracks of the disk in the event that there is a sudden incident during the remote loading operation. The JEEP module also contains a reboot manager that is responsible for changing the execution level of the files of the jukebox once the automatic file manager has determined that an update of the jukebox has been accomplished. The JEEP.DLL module also generates a MISDEPS.DAT file when a missing dependence has been detected. This file contains lines in the form NEEDPARSER.DLL arrow version 2: NEEDLEXY.DLL -> version 2.0, etc.... This file allows the server, by reading this MISDEPS.DAT file, to determine the modules on the jukebox and to reload them.

Other modifications within the grasp of one skilled in the art are also part of the spirit of the invention.

## Claims

1. System for remote loading of objects or files in order to update software, wherein the architecture of the operating system calls for the different tasks to be broken down into software modules that are connected to one another by dynamic links or consist of executable subroutines with primary dependence with other parts of the operating system, whereby each of the modules consists of object files or libraries that are represented by dynamic link libraries, which are organized among themselves according to a number of dependence levels that are described in their respective attributes.

2. System for remote loading of objects or files in order to update software according to claim 1, wherein the attributes of an object or a library indicate the version number and the dependencies of the object with respect to other objects.

3. System for remote loading of objects or files in order to update software according to claim 2, wherein the attributes indicate the levels attributed to the modules.

4. System for remote loading of objects or files in order to update software according to claims 1-3, wherein the different tasks include a main task, which includes a module (JUKECORE), whose purpose is to load the dynamic link libraries (DLL), to initialize the nucleus of the program, to initialize the graphics management module (GFX), to initialize the library loading module (WDLL), to load the Telecom module of the telecommunications tasks (TELECOM.DLL), and to launch the screen interpreter program

as a main task.

5. System for remote loading of objects or files in order to update software according to claim 4, wherein the program is subdivided into a certain number of modules that define a task that is specific to the terminal.

6. System for remote loading of objects or files in order to update software according to claim 5, wherein said specific task is that which corresponds to a jukebox and includes a specific main task, a certain number of modules that define the "Windows" functions of the display, which are the following:

- a module for running the mouse or the touch-screen signals;

- a module for running the messages exchanged among the objects;

- a FIL.DJL module for managing the files on disk;

- a FILIO.DJL read-write module for files on disk;

- a JSTRUCT.DJL file for monitoring all of the events generated by the hardware, such as the touch screen, the sound card, and the money hardware processing interface.

7. System for remote loading of objects or files in order to update software according to claim 5 or 6, wherein the specific main task of the jukebox application includes a module (SILOAD.DLL) that contains the loader program library whose purpose is to verify the required dynamic link library versions, to load them, and to call the Telecom task module in order to transfer files.

8. System for remote loading of objects or files in order to update software according to claim 7, wherein said SILOAD module contains the list in a file (DLL.DEFAULT) of the minimum versions that are acquired for operation, as well as the list of all of the functions that are represented either by libraries (DATA JUKEBOX LIBRARY) or by object files (DJO, DATA JUKEBOX OBJECT).

9. System for remote loading of objects or files in order to update software according to claim 8, wherein each object or library contains the list of all of the functions that the library or object needs, as well as the version numbers and dependencies.

10. System for remote loading of objects or files in order to update software according to claim 4, wherein WDLL manages all of the new modules and verifies that the remotely loaded modules have no missing dependencies and have been loaded with the necessary versions.

11. System for remote loading of objects or files in order to update software according to claim 7, wherein SILOAD manages the loading of the modules that are specific to the task of the terminal, i.e., all of the DJL modules already listed above, as well as the jukebox library modules consisting of WOBJECT which manages the objects, the mixer, and purchases; the WCURSOR module, which manages the movements of the cursor; the DBMAPI module, which manages the database; the WFONTS module, which manages all the font types; the PARSER module (PARSER analysis

program), which analyzes and generates the screens from the script and verifies the grammar with the aid of the "GRAMMAR.DJL" module and the lexical module "LEXY.DJL."

12. System for remote loading of objects or files in order to update software according to claim 11, wherein the library loading module SILOAD includes a "WINDEF" module, which contains the list of the files that have to be included in order to manage the windows of a Windows display that is provided on the monitor of the jukebox-type terminal.

13. System for remote loading of objects or files in order to update software according to claim 12, wherein said list of objects consist of:

- an object module "OBJET WPSCREEN.DJO", which makes it possible to define the main page on the monitor;
- a "WSCREEN.DJO" module, which makes it possible to determine on this main page the number of screens available and thus to make it possible to display multiple windows or screens;
- a "WIMAGE.DJO" module, which makes it possible to determine and define on the screen the image that the screen will use;
- a "WANIM.DJO" module, which makes it possible to define the animation when the image is animated;
- a "WBUTTON.DJO" module, which makes it possible to define and manage the buttons that are used on the screen of the main page;

- a "WLABEL.DJO" module, which makes it possible to create labels that make it possible to write on top of an object; and
- a "WSCROLLER.DJO" module, which makes it possible to define scrolling display zones, for instance between two points, vertical, horizontal, diagonal.

14. System for remote loading of objects or files in order to update software according to claim 13, wherein all of these object modules that are managed by the main task use a "JHANDLER" library, which makes it possible to define the fixed uses of the screens and thus to determine which are the interfaces that ensure the links to the different objects that are defined by the previous modules.

15. System for remote loading of objects or files in order to update software according to claim 7, wherein the SILOAD task launches or loads the "XCP" module, making it possible to manage payment tasks such as ticket receiving systems or coin or card payment units and also makes it possible to back up the basic information in the IBUTTON.



## **Abstract**

Inventor: Guy Nathan

Dominique Dion

Filed by: TECHNICAL MAINTENANCE CORP. INC.

This invention pertains to a system for remote loading of objects or files in order to update software.

Said system for remote loading of objects or files to update software is characterized in that the architecture of the operating system calls for the different tasks to be broken down into software modules that are connected to another by dynamic links or are composed of executable subroutines that have main dependence links to other parts of the operating system, whereby each of the modules is composed of object files or libraries that are represented by dynamic link libraries among themselves according to a number of dependence levels that are described in their respective attributes.

Figure 2

[in Fig. 2:  
order in which dynamic links are loaded  
main dependencies  
module  
task]

Translator's notes:

1. The term "auto-alimenté" pertaining to static RAM (p. 6 of the foreign, p. 4 of the English), was not found in available references and was thus rendered literally as "self-powered," but it may refer to the fact that such RAM is volatile, i.e., it will lose data if it has no power.

2. The paragraph starting "A high-speed telecommunications modem adapter..." (foreign page 6, English p. 5) is an incomplete sentence.

3. The second sentence in the paragraph starting "By means of its input controller circuit (3),..." (foreign page 6, English p. 5) is an incomplete.

PL 1/2

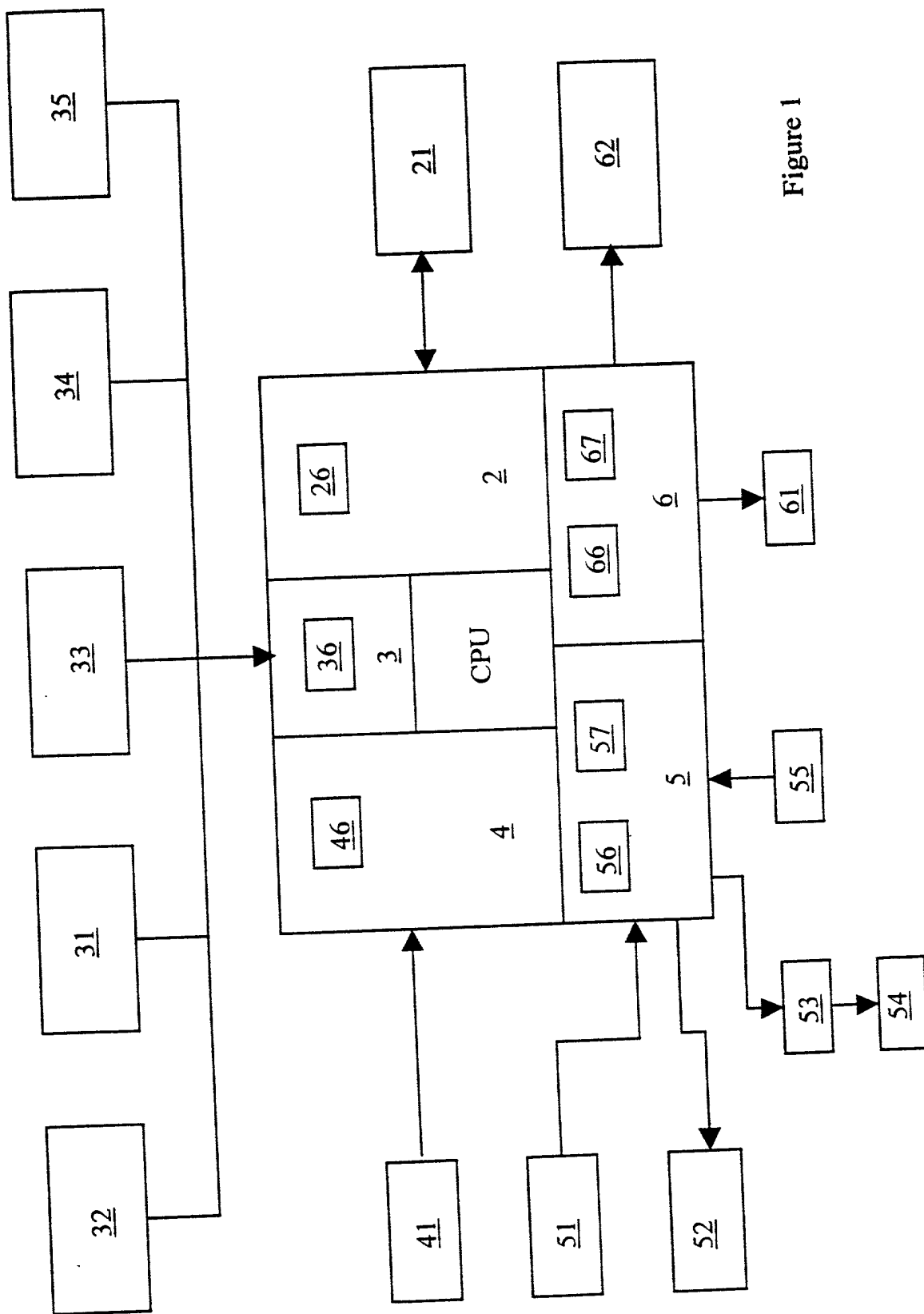
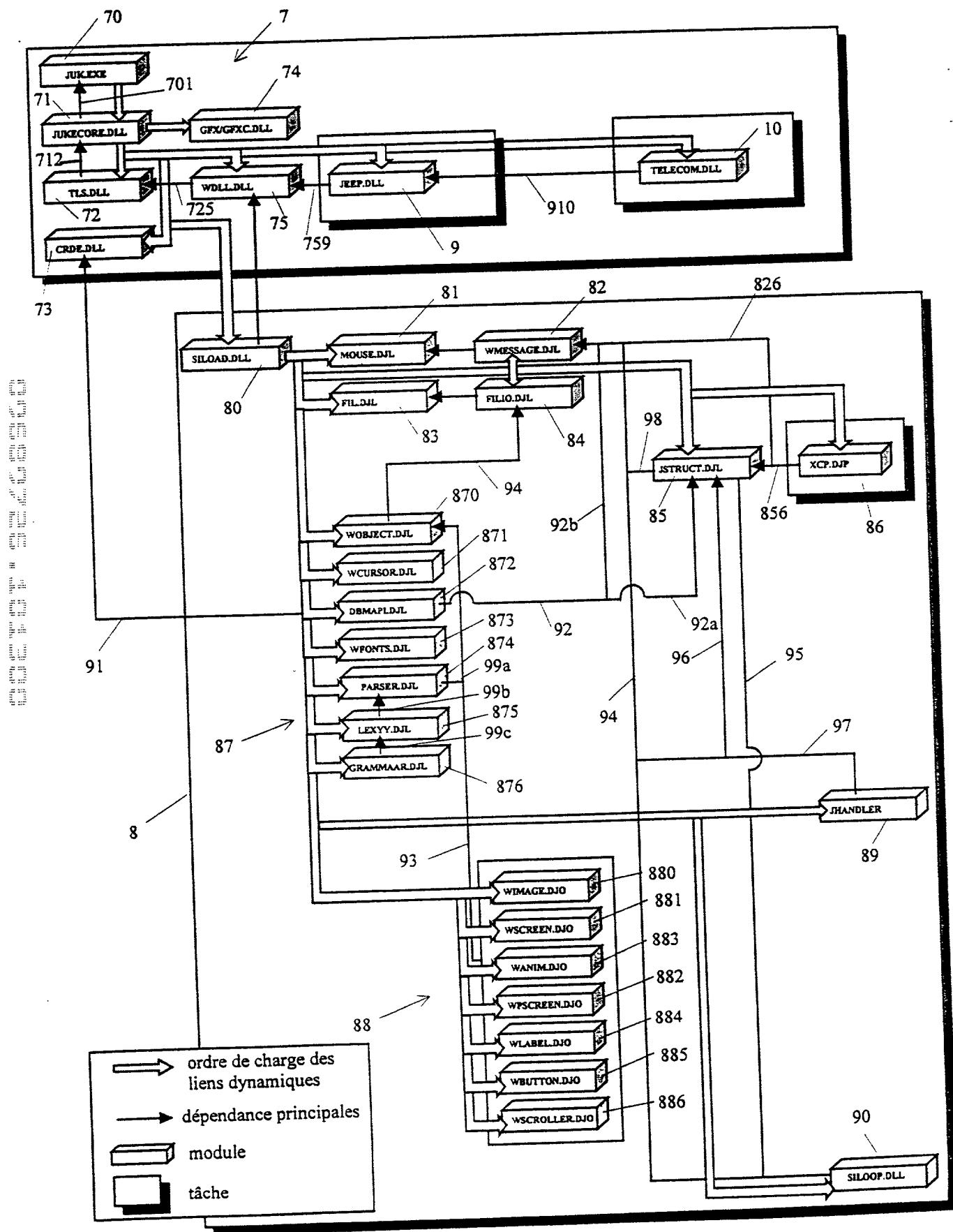


Figure 1

Figure 2



**RULE 63 (37 C.F.R. 1.63)**  
**DECLARATION AND POWER OF ATTORNEY**  
**FOR PATENT APPLICATION**  
**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

I, a below named inventor, I hereby declare that my residence, post office address and citizenship are as stated below to my name, and I believe I the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

**SYSTEM FOR REMOTE LOADING OF OBJECTS OR FILES IN ORDER TO UPDATE SOFTWARE**

the specification of which (check applicable box(s)):

- ☐ is attached hereto  
☐ was filed on September 1, 1998 as U.S. Application Serial N°. (To be assigned) (Atty Dkt. N) 871-49)  
☐ was filed as PCT International application N°. \_\_\_\_\_ on \_\_\_\_\_  
and (if applicable to U.S. or PCT application) was amended on \_\_\_\_\_

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above. I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with 37 C.F.R. 1.56. I hereby claim foreign priority benefits under 35 U.S.C. 119/365 of any foreign application (s) for patent or inventor's certificate which priority is claimed or, if no priority is claimed, before the filing date of the application:

Priority Foreign Application(s):

Application Number	Country	Day/Month/Year
Filed		
<u>98 09296</u>	<u>FRANCE</u>	<u>21 July 1998</u>

I hereby claim the benefit under 35 U.S.C. §119(e) of any United States provisional application(s) listed below.

Application Number	Date/Month/Year Filed
--------------------	-----------------------

I hereby claim the benefit under 35 U.S.C. 120/365 of all prior United States and PCT international applications listed above or below and, insofar as the subject matter of each of the claims of this application is not disclosed in such prior applications in the manner provided by the first paragraph of 35 U.S.C. 112, I acknowledge the duty to disclose material information as defined in 37 C.F.R. 1.56 which occurred between the filing date of the prior applications and the national or PCT international filing date of this application:

Prior U.S./PCT Application(s):

Application Serial N°	Date/Month/Year Filed	Status: patented pending, abandoned
-----------------------	-----------------------	--

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon. And I hereby appoint **NIXON & VANDERHYE P.C., 1100 North Glebe Rd., 8<sup>th</sup> Floor, Arlington, VA 22201-4714, telephone number (703) 816-4000 (to whom all communications are to be directed)**, and the following attorneys thereof (of the same address) individually and collectively my attorneys to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith and with the resulting patent: Arthur R. Crawford, 25327; Larry S. Nixon, 25640; Robert A. Vanderhye, 27076; James T. Hosmer, 30184; Robert W. Faris, 31352; Richard G. Besho, 22770; Mark E. Nusbaum, 32348; Michael J. Keenan, 32106; Bryan H. Davidson, 30251; Stanley C. Spooner, 27393; Leonard C. Mitchard, 29009; Duane M. Byers, 33363; Jeffrey H. Nelson, 30481; John R. Lastova, 33149; H. Warren Burnam, Jr. 29366; Thomas E. Byrne, 32205; Mary J. Wilson, 32955; J. Scott Davidson, 33489; Alan M. Kagen, 36178; William J. Griffin, 31260; Robert A. Molan, 29834; B.J. Sadoff, 36663; James D. Berquist, 34776; Updeep S. Gill, 37334.\*

- Inventor's Signature: \_\_\_\_\_ Date: September 22, 1998  
Inventor: Guy NATHAN  
(first) MI (last) French (citizenship)  
Residence: (city) YERRES (state/country) FRANCE  
Post Office Address: 1, rue Jeanne d'Arc, 91330 YERRES, France  
(Zip Code): \_\_\_\_\_
- Inventor's Signature: \_\_\_\_\_ Date: The 22<sup>th</sup> of September 1998  
Inventor: Dominique DION  
(first) MI (last) Canadian (citizenship)  
Residence: (city) LAVAL (state/country) Quebec, CANADA  
Post Office Address: 411, Yvon Berger, Laval, Quebec, Canada  
(Zip Code): H7L 4N8
- Inventor's Signature: \_\_\_\_\_ Date: \_\_\_\_\_  
Inventor: \_\_\_\_\_ MI \_\_\_\_\_  
(first) (last) (citizenship)  
Residence: (city) \_\_\_\_\_ (state/country) \_\_\_\_\_  
Post Office Address: \_\_\_\_\_  
(Zip Code): \_\_\_\_\_

## **SYSTEME DE TELECHARGEMENT D'OBJETS OU DE FICHIERS POUR** **MISE A JOUR DE LOGICIELS**

La présente invention concerne un système de téléchargement  
5 d'objets ou de fichiers pour mise à jour de logiciels, notamment pour les  
systèmes de reproduction audiovisuelle déclenchés par paiement de  
redevances, tels que juke-box ou autres appareils.

Il est connu dans l'art antérieur des dispositifs de téléchargement de  
système d'exploitation à travers un réseau tels que, par exemple par la  
10 demande de brevet britannique N° 2 231 180. Cette demande de brevet  
enseigne d'effectuer le chargement d'un premier ordinateur lié à travers un  
réseau à un deuxième ordinateur en chargeant un sous-ensemble du  
système d'exploitation dans la mémoire du premier ordinateur, le sous-  
ensemble comprenant les commandes permettant la copie des fichiers, la  
15 création d'un répertoire, le formatage du disque ainsi que le fonctionnement  
d'une connexion à travers le réseau pour ensuite utiliser ce sous-ensemble  
pour transférer du second ordinateur l'ensemble des fichiers de système  
d'exploitation sur le disque du premier ordinateur.

Dans ce type de téléchargement, il convient d'effectuer le  
20 chargement de l'ensemble du système d'exploitation et de tous les fichiers  
du système d'exploitation. Ceci amène donc soit à limiter les  
téléchargements, soit à occuper, pendant des temps relativement longs, des  
systèmes de télécommunications pour effectuer des téléchargements dans  
le cas où les systèmes d'exploitation ou les fichiers associés doivent être  
25 renouvelés fréquemment.

Il est également connu par le brevet américain N° 4 958 278 un  
système de téléchargement sur un terminal non pourvu de lecteur de  
disques.

Il est également connu par le brevet français N° 2 682 786 un autre procédé de téléchargement sur un terminal non pourvu de disque dur.

Enfin un dernier brevet européen N° 0 358 992 enseigne une méthode de téléchargement à travers un réseau de système d'exploitation  
5 ou de programmes exécutables sur un ordinateur qui ne comporte pas de dispositif d'amorçage ou d'autres dispositifs contenant le programme exécutable. Un premier programme d'amorçage minimum est transféré dans un premier temps, puis ce premier programme d'amorçage minimum s'exécute, établit une liaison logique vers un disque du serveur et autorise  
10 l'ordinateur de requête à traiter le disque du serveur comme un dispositif d'amorçage local.

Le but de l'invention est d'éviter la nécessité d'une part de réamorcer le terminal sur lequel on télécharge et d'autre part, de permettre de transférer des fichiers d'exploitation ou des parties de programme  
15 exécutable sans avoir à réinitialiser la machine, tout en étant sûr que le fonctionnement du système ne sera pas dégradé par la version téléchargée.

Ce but est atteint par le fait que l'architecture du système d'exploitation comporte une répartition des différentes tâches en modules logiciels reliés entre eux par des liens dynamiques ou constitués de sous-  
20 programmes exécutables présentant des liens de dépendance principale avec d'autres parties du système d'exploitation, chacun des modules étant constitué de fichiers d'objets ou de bibliothèques représentées par des bibliothèques de liens dynamiques "dynamic links library" organisées entre elles selon une pluralité de niveaux de dépendance décrits dans leurs attributs respectifs.

25 Selon une autre particularité, les attributs d'un objet ou d'une bibliothèque indiquent le numéro de version et les dépendances de l'objet par rapport à d'autres objets.

Selon une autre particularité, les attributs indiquent le niveau attribué aux modules.



Selon une autre particularité, les différentes tâches comprennent une tâche principale qui comporte un module (JUKECORE) destiné à charger les librairies de liens dynamiques (DLL), à initialiser le noyau du programme, à initialiser le module de gestion des graphiques (GFX), à  
 5 initialiser le module de chargement des librairies (WDLL), à charger le module Télécom des tâches de télécommunications (TELECOM.DLL) et à lancer le programme interpréteur d'écran en tant que tâche principale.

Selon une autre particularité, le programme est subdivisé en un certain nombre de modules définissant une tâche spécifique au terminal.

10 Selon une autre particularité, cette tâche spécifique est celle correspondant à un juke-box et comporte une tâche principale spécifique, un certain nombre de modules définissant les fonctions "fenêtres" de l'affichage, qui sont les suivantes :

- un module d'exploitation des signaux souris ou de l'écran tactile ;
- 15 - un module d'exploitation des messages échangés entre les objets ;
- un module "FIL.DJL" de gestion des fichiers sur disques ;
- un module "FILIO.DJL" de lecture-écriture de fichiers sur disques ;
- un module "JSTRUCT.DJL" de surveillance de tous les événements générés par les équipements, tels que l'écran tactile, la carte  
 20 son, l'interface de traitement des équipements monétaire.

Selon une autre particularité, la tâche principale de l'application juke-box comporte un module "SILOAD.DLL" qui contient la librairie du programme chargeur destinée à vérifier les versions de librairies de liens dynamiques requises, de les charger et d'appeler le module des tâches  
 25 Télécom pour effectuer le transfert de fichiers.

Selon une autre particularité, ce module SILOAD comporte la liste dans un fichier (DLL.DEFAULT) des versions minimales nécessaires au fonctionnement ainsi que la liste de toutes les fonctions représentées soit par des bibliothèques (DJL) (DATA JUKEBOX LIBRARY), soit par des  
 30 fichiers d'objets (DJO DATA JUKEBOX OBJECT).

Selon une autre particularité, chaque objet ou bibliothèque contient la liste de toutes les fonctions, dont la bibliothèque ou l'objet a besoin, ainsi que les numéros de version et les dépendances.

5 Selon une autre particularité, WDLL fait la gestion de tous les nouveaux modules, vérifie que les modules téléchargés n'ont aucune dépendance manquante et ont été chargés avec les versions nécessaires.

Selon une autre particularité, SILOAD gère le chargement, des modules spécifiques à la tâche du terminal, à savoir tous les modules "DJL" déjà énumérés ainsi que les modules de librairie juke-box constitués par  
10 WOBJECT gèrent les objets, le mixeur, les achats ; le module "WCURSOR" gère les mouvements du curseur ; le module DBMAPI gère la base de donnée ; le module "WFORMATS" gère tous les types de fonte ; le module "PARSER" (PARSER Programme d'analyse) analyse et génère les écrans à partir du script et vérifie la grammaire à l'aide du module "GRAMMAR.DJL"  
15 et le module lexique "LEXY.DJL".

Selon une autre particularité, le module de chargement de librairie SILOAD comporte un module "WINDEF" qui contient la liste des fichiers qu'il faut inclure pour gérer les fenêtres d'un affichage fenêtres prévu sur le moniteur du terminal de type juke-box.

20 Selon une autre particularité, cette liste d'objets est constituée :

- d'un module "OBJET WPSCREEN.DJO" qui permet de définir la page principale sur le moniteur ;

- d'un module objet "WSCREEN.DJO" qui permet de déterminer dans cette page principale le nombre d'écrans disponibles et ainsi permettre  
25 l'affichage de plusieurs fenêtres ou écrans ;

- d'un module "WIMAGE.DJO" qui permet de déterminer et de définir dans l'écran l'image qu'il utilisera ;

- d'un module "WANIM.DJO" qui permet de définir l'animation lorsque l'image est animée ;

- d'un module "WBUTTON.DJO" qui permet de définir et gérer les  
30 boutons qui sont utilisés sur l'écran de la page principale ;

- d'un module "WLABEL.DJO" qui permet de créer des étiquettes permettant l'écriture par-dessus un objet ; et

- d'un module "WSCROLLER.DJO" qui permet de définir des zones d'affichage à défilement, entre deux points par exemple, horizontal, diagonal, vertical.

Selon une autre particularité, l'ensemble de ces modules d'objets, gérés par la tâche principale, utilise une librairie "JHANDLER" qui permet de définir les utilisations fixes des écrans et donc, de déterminer quelles sont les interfaces qui assurent la liaison avec les différents objets définis par les modules précédents.

Selon une autre particularité, la tâche SILOAD lance ou charge le module "XCP", permet de gérer les tâches de paiement telles que les systèmes de réception de billets ou les unités de paiement à monnaie ou à carte et permet également la sauvegarde des informations de base dans le IBUTTON.

D'autres particularités et avantages de la présente invention apparaîtront plus clairement à la lecture de la description ci-après faite en référence aux dessins annexés dans lesquels :

la figure 1 représente une vue schématique du schéma électrique du matériel constituant l'intention,

la figure 2 représente un logigramme des relations entre les modules de librairies et les modules d'objets.

De préférence mais cependant de manière non limitative, le système de reproduction audiovisuelle utilise les éléments matériels énumérés et référencés ci-après.

L'unité centrale 1 à microprocesseur est un système compatible PC de haute performance, le choix lors de la mise en oeuvre s'étant porté sur un système du type Pentium qui possède des moyens de mémorisation et les caractéristiques suivantes:

- compatibilité avec le bus local Vesa,
- antémémoire de processeur: 256 kO minimum,
- mémoire vive: RAM de 32 MO
- ports série de haute performance,
- 5 - adaptateur graphique type SVGA à microprocesseur
- contrôleur de bus type SCSI/2,
- mémoire vive RAM statique auto-alimentée.

Tout autre unité centrale, possédant des performances équivalentes ou supérieures, pourra être utilisée dans l'invention.

10 Cette unité centrale commande et gère une circuit de commande de son (5), un circuit (4) de commande des télécommunications, un circuit (3) de commande des entrées, un circuit (2) de commande de la mémorisation de masse et un circuit (6) de commande des moyens de visualisation. Les moyens de visualisation se composent principalement d'un moniteur vidéo

15 (62) à écran plat sans entrelacement de type SVGA à haute résolution et faible rayonnement, c'est ce moniteur qui est utilisé pour la reproduction d'images (par exemple les couvertures d'albums des sélections musicales), de graphiques ou de clips vidéo,

Des moyens de mémorisation de masse (21), utilisant des disques

20 durs de type SCSI haute vitesse et haute capacité, sont associés aux moyens de mémorisation déjà présents dans le dispositif à microprocesseur. Ces moyens servent au stockage d'informations audiovisuelles numérisées et compressées.

Un adaptateur de modem de télécommunications (41) haute vitesse

25 qui est intégré pour autoriser la liaison avec un réseau de distribution d'informations audiovisuelles contrôlé par un serveur central.

Pour la reproduction des informations sonores des sélections musicales, le système comporte des haut-parleurs (54) recevant le signal

d'un amplificateur-tuner (53) relié au circuit électronique (5) incorporant deux tampons mémoire (56, 57) et de type synthétiseur de musique prévu pour supporter un grand nombre de sources d'entrée tout en fournissant une sortie présentant une qualité de type CD (disque compact).

- 5 De même le circuit de commande des moyens de visualisation comporte également deux mémoires tampon (66, 67) dans le but explicité ultérieurement.

Une alimentation thermiquement régulée de 240 watts ventilée fournit l'énergie au système. Cette alimentation est protégée contre les  
10 surintensités et les suroscillations.

Le système de reproduction audiovisuelle gère, par le biais de son circuit contrôleur d'entrée (3), un écran tactile (33), qui inclut un panneau de revêtement de la vitre utilisant la "technologie avancée d'onde de surface" ainsi qu'un contrôleur de bus de type AT. Cet écran tactile permet après  
15 avoir affiché sur le moniteur vidéo (62) ou l'écran d'un téléviseur (61) diverses informations de sélection utilisées par les clients, ainsi que des informations de commandes et de contrôle de gestion utilisées par le gérant ou le propriétaire du système. Il est également utilisé à des fins de maintenance en combinaison avec un clavier externe (34) qui peut être relié  
20 au système qui possède pour cela un connecteur de clavier, contrôlé par un verrou à clé (32) à travers le circuit d'interface (3).

Le circuit d'entrée (3) interface également le système avec un ensemble télécommande (31) constitué, par exemple, d'une télécommande RF à fréquence à radio.

- 25 Un dispositif de paiement de redevances (35) est également relié au circuit d'interface d'entrée (3). Il est également possible d'utiliser tout autre dispositif qui permette la réception de tout mode de paiement par pièces, billets, jetons, cartes magnétiques ou à puces ou toute combinaison des moyens de paiement

Pour loger le système, il est de plus prévu un châssis ou bâti avec garnitures extérieures personnalisables.

Outre ces éléments, un microphone (55) sans fil est relié au contrôleur de son (5), ce qui permet de transformer ce dernier en un puissant système d'annonces et d'informations destinées au public ou éventuellement en machine de karaoké. De même un système de haut-parleurs sans fil peut être utilisé par le système.

L'ensemble (31) de commande à distance permet au gérant, par exemple de derrière le bar, d'accéder et de contrôler différentes commandes telles que:

- la commande marche/arrêt du microphone,
- la commande de mise en sourdine des haut-parleurs,
- la commande de contrôle de volume sonore,
- la commande d'annulation de la sélection musicale en train d'être écoutée.

Deux tampons (56, 57) sont associés au circuit contrôleur de son (5) pour permettre de mémoriser chacun de façon alternative une information correspondant à environ au moins un quart de seconde de son. De même deux tampons (66, 67) sont associés au circuit contrôleur de vidéo (6) capables chacun et alternativement de mémoriser environ au moins un dixième de seconde d'images. Enfin un tampon respectif (46, 36, 26) est associé à chacun des circuits contrôleur de communication(4), d'interface d'entrée (3) et de mémorisation (2).

Le logiciel d'exploitation du système a été élaboré autour d'une librairie d'outils et de services très largement orientés vers le domaine audiovisuel dans un univers multimédia. Cette librairie inclut de manière avantageuse un système d'exploitation multitâche performant qui autorise efficacement l'exécution simultanée de multiples fragments de code. Ce logiciel d'exploitation permet ainsi les exécutions concurrentes, de manière

ordonnée et en évitant tout conflit, d'opérations réalisées sur les moyens de visualisation, les moyens de reproduction sonores de même que la gestion des liaisons de télécommunications au travers du réseau de distribution. De plus, ce logiciel présente une grande flexibilité

5 Le système d'exploitation est partagé en modules comportant un premier module d'amorçage (7) lui-même subdivisé en un premier module (70) de programme principal "JUK.EXE" qui vérifie la mémoire et vérifie si le minimum d'objets est présent pour assurer le fonctionnement du juke-box ; un second module (71) lié dynamiquement et dépendant de celui-ci est  
10 constitué par le module "JUKECORE.DLL". La fonction de ce second module (71) est de contenir les librairies de langage C et d'assurer la tâche principale.

L'architecture du système d'exploitation comporte une répartition des différentes tâches en modules logiciels reliés entre eux par des liens  
15 dynamiques ou constitués de sous-programmes exécutables présentant des liens de dépendance principale avec d'autres parties du système d'exploitation. Chacun des modules est constitué de fichiers d'objets ou de librairies de liens dynamiques "dynamic links library" organisés selon une pluralité de niveaux de dépendance décrits dans les attributs. Les attributs  
20 d'un objet ou d'une librairie indiquent le numéro de version et les dépendances du fichier d'objet ou de la librairie par rapport à d'autres fichiers d'objets, comme décrit ci-après pour le module PARSE. Chaque attribut indique le niveau attribué au module. Ainsi, le module JUK.EXE (70) est de niveau supérieur aux modules JUKECORE (71), TLS (72), CRDE  
25 (73), GFX (74), WDLL (75), JEEP (9) et TELECOM (10), mais le module TELECOM (10) est dépendant du module JEEP (9) (voir lien 910), donc de niveau inférieur à JEEP (9).

De même, JEEP (9) (voir lien 759) est de niveau inférieur au module WDLL (75), car dépendant de celui-ci et TLS (lien 725) de niveau supérieur  
30 à WDLL (75). Par contre, TLS et GFX peuvent être de même niveau. La

tâche principale comporte un module (JUKECORE) destinée à initialiser ou charger le module (73), le noyau du programme "CRDE.DLL", à initialiser ou charger le module (74) de gestion des graphiques (GFX), à initialiser ou charger le module (75) de chargement (WDLL.DLL) des librairies, à charger le module (10) Télécom des tâches de télécommunications (DLL), à charger le module (72) TLS.DLL qui contient toutes les utilisations nécessaires au juke-box, aux télécoms, heure, décryptage, etc....., à initialiser ou charger la librairie des programmes JEEP (Juke Execution Exchange Protocol) assurant les tâches de serveur d'intégrité, de requête de chargement et le dialogue avec le serveur, et à lancer le programme (80, SILOAD.DLL) en tant que tâche principale. La tâche principale de l'application juke-box comporte un module (SILOAD.DLL) contenant la librairie du programme chargeur destinée à vérifier les versions de librairie à liens dynamiques requises dans (WDLL), de les charger ou d'appeler le module des tâches, Télécom pour effectuer le transfert de fichiers. Le module WDLL DLL comporte la liste dans un fichier (DLL.DEFAULT) des versions minimales nécessaires au fonctionnement ainsi que la liste de toutes les fonctions représentées soit par des bibliothèques (LIBRARY) (DLL, DJL), soit par des fichiers d'objets (DJO). Chaque objet ou bibliothèque contient la liste de toutes les fonctions, dont la bibliothèque ou l'objet a besoin, ainsi que les numéros de version et les dépendances. Le module WDLL fait la gestion de tous les nouveaux module, s'assure des interdépendances et vérifie que les modules téléchargés n'ont aucune dépendance et ont été chargés avec les versions nécessaires. La partie applicative (8) propre à un juke-box comporte un certain nombre de modules chargés et lancés par SILOAD et définissant les fonctions "fenêtres" de l'affichage, qui sont les suivantes :

- un module (81) d'exploitation des signaux souris ou de l'écran tactile ;
- un module (82) d'exploitation des messages échangés entre les objets et les différents autres modules ;
- un module (83) FIL.DJL de gestion des fichiers sur disques ;



- un module (84) FILIO.DJL de lecture-écriture de fichiers sur disques ;

- un module (85) JSTRUCT.DJL de surveillance de tous les événements générés par les équipements, tels que l'écran tactile, la carte son, l'interface de traitement des équipements monétaire.

SILOAD gère le chargement, des modules spécifiques à la tâche du terminal, à savoir tous les modules DJL déjà énumérés ainsi que les modules de librairie juke-box (87) constitués par WOBJECT (870) qui gère les objets, tels que le mixeur, les achats ; le module WCURSOR (871) qui gère les mouvements du curseur ; le module (872) DBMAPI qui gère la base de donnée ; le module (873) WFONTs qui gère tous les types de fonte ; le module (874) PARSEr (Programme d'analyse syntaxique) qui analyse et génère les écrans à partir du script et vérifie la grammaire à l'aide du module (876) "GRAMMAR.DJL" et le module (875) "LEXY.DJL" qui est le module lexique d'assignation des fonctions des mots à l'intérieur du langage. Le module PARSEr (874) contient dans son en-tête de fichier les informations suivantes :

```
- char *parser_version_info= "DLL_INFO DJL;"
                                "DLL-NAME PARSEr.DJL;"
                                "VERSION 1;"
                                "CREATOR KENDALF;"
                                "REQUIRIES lexyy.djl;4;"
                                "REQUIRIES grammar-.djl;5";
```

Tous les modules et toutes les librairies (DJO, DLL, DJL) contiennent des informations semblables à celles du module PARSEr et ces informations déterminent les nécessités de version et de dépendance.

Ainsi, le module PARSEr a besoin des modules LEXY version 4 et GRAMMAR version 5 pour permettre son exécution par le système. Les flèches doubles de la figure 2, reliant les différents modules entre eux, donnent l'ordre de chargement des différents fichiers. Ainsi, comme on l'a vu précédemment, il faudra commencer par charger JUKE.EXE pour ensuite charger JUKECORE.DLL et pouvoir, à partir de JUKECORE.DLL, charger

GFX.DLL, TLS.DLL, WDLL.DLL, JEEP.DLL, TELECOM.DLL, CRDE.DLL et SILOAD.DLL.

Les flèches uniques représentent les dépendances entre fichiers. Ainsi, la flèche (91) indique que les fichiers DJL et en particulier, DBMAPI sont dépendants de CRDE.DLL. La flèche (93) montre que les fichiers DJO sont dépendants du module WOBJECT.DJL. Le module WOBJECT.DJL est lui-même dépendant du module FILIO.DJL. La flèche (92a) indique que DBMAPI.DJL est dépendant de JSTRUCT.DJL et la flèche (92b) que DBMAPI.DJL est dépendant de WMESSAGE.DJL. La flèche (98) indique que JSTRUCT.DJL est dépendant par le fichier WMESSAGE.DJL. WMESSAGE est dépendant du fichier MOUSE.DJL et FILIO.DJL étant dépendant du fichier FIL.DJL. le fichier XCP.DJL est dépendant, par la flèche (856), de JSTRUCT.DJL et comme insiqué, par la flèche (826) de WMESSAGE.DJL. Le fichier JHANDLER est dépendant, par la flèche (97), de WMESSAGE.DJL et comme représenté, par la flèche (96) de JSTRUCT.DJL. Le fichier SILOOP.DLL est dépendant, comme représenté par la flèche (95), de JSTRUCT.DJL et par la flèche (94) de WMESSAGE.DJL. Le fichier TELECOM.DLL est dépendant, comme indiqué par la flèche (910), de JEEP.DLL qui est lui-même dépendant, comme montré par la flèche (959), de WDLL.DLL. Le fichier WDLL.DLL est dépendant, par la flèche (725), de TLS.DLL. De même, la flèche (89c) montre que GRAMMAR.DJL est dépendant de LEXY.DLL et, par la flèche (99b), que LEXY.DJL est dépendant de PARSE.DJLL. Ainsi, comme on a pu le voir précédemment, PARSE a besoin de LEXY et de GRAMMAR pour s'exécuter et la version 1 de PARSE fait appel à la version 4 de LEXY.DJL et à la version 5 de GRAMMAR.DJL. De même, WOBJECT.DJL est dépendant par la flèche (99a) de PARSE.DJL. Ainsi, tous les modules et toutes les librairies .DJO, .DLL, .DJL contiennent des informations semblables à celles du module PARSE qui déterminent les nécessités de version des différents modules dont un module donné est dépendant. Ces

informations donnent également les dépendances du module vis-à-vis des autres modules, comme cela est indiqué par les flèches sur la figure 2.

Le module de chargement de librairie SILOAD charge également ou lance un module SILOOP.DLL (90) qui est une bande d'attente d'événement.

- 5 Un ensemble de modules (88) contient la liste des fichiers qu'il faut inclure pour gérer les fenêtres d'un affichage fenêtres prévu sur le moniteur du terminal de type juke-box.

Cette liste d'objets est constituée :

- d'un fichier d'objets (883) "WPSCREEN.DJO" qui permet de définir  
10 la page principale sur le moniteur ;
- d'un fichier d'objets (881) "WSCREEN" qui permet de déterminer dans cette page principale le nombre d'écrans disponibles et ainsi permettre l'affichage de plusieurs fenêtres ou écrans ;
- d'un fichier d'objets (880) "WIMAGE.DJO" qui permet de  
15 déterminer et de définir dans l'écran l'image qu'il utilisera ;
- d'un fichier d'objets (882) "WANIM.DJO" qui permet de définir l'animation lorsque l'image est animée ;
- d'un fichier d'objets (885) "WBUTTON.DJO" qui permet de définir et  
20 gérer les boutons qui sont utilisés sur l'écran de la page principale tels que les boutons d'actionnement utilisés dans l'interface graphique définie dans la demande de brevet PCT WO 96/12258 ;
- d'un fichier d'objets (884) "WLABEL.DJO" qui permet de créer des étiquettes permettant l'écriture par-dessus un objet ; et
- d'un fichier d'objets (886) "WSCROLLER.DJO" qui permet de  
25 définir des zones d'affichage à défilement vertical.

Une librairie "JHANDLER" permet de définir les utilisations fixes des écrans et donc, de déterminer quelles sont les interfaces qui assurent la liaison avec les différents objets définis par les modules précédents.

- 30 Le module (86) de librairie "XCP" permet de gérer les tâches de paiement telles que les systèmes de réception de billets ou les unités de paiement à monnaie ou à carte et permet également la sauvegarde des

informations de base dans le IBUTTON qui est un circuit intégré de mémorisation de codes secrets pour l'utilisateur.

Ainsi, lorsqu'un nouveau fichier est envoyé par téléchargement au système, le fichier comporte des informations sur son niveau qui dépend du type de fichier. Les fichiers des images graphiques, par exemple WIMAGE.GJO, ont les niveaux les plus élevés et les fichiers de gestion des parties Hardware, tels que, par exemple XCP.DJL, ont les niveaux les plus bas. Le module JEEP vérifie la logique de dépendance en commençant par les fichiers de niveau les plus bas et en remontant vers les fichiers supérieurs tout en s'assurant que les dépendances nécessaires entre les fichiers ou librairies sont respectées. De cette façon, une modification dans WOBJECT.DJL va amener JEEP à vérifier que les informations de version nécessaires à WOBJECT.DJL pour les fichiers DJO dépendants et nécessaires à son exécution correspondent aux versions minimales requises par les informations inscrites dans le fichier WOBJECT.DJL. Ainsi, si WOBJECT.DJL nécessite une version déterminée de WPSCREEN.DJO, il vérifiera que cette version est au minimum présente et s'il n'y a qu'une version de niveau insuffisant, il signalera un problème. Puis JEEP va remonter les liens de dépendance vers FILIO.DJL et FIL.DJL.

L'organisation du disque dur du Juke box est effectuée de façon à comporter un répertoire C:\NEWJUKE qui contient les nouveaux fichiers Juke box lorsqu'on télécharge de nouveaux modules. Un autre fichier C:\OLDJUKE contient la sauvegarde des versions stables des fichiers et modules nécessaires au fonctionnement du Juke box. Le module JEEP (JUKE EXECUTION EXCHANGE PROTOCOL) contient un gestionnaire automatique de fichiers qui conserve la trace des modules et fichiers mis à jour en sauvegardant les fichiers anciens pendant un certain temps et en déplaçant les nouveaux fichiers dans le répertoire NEWJUKE. Cette tâche enregistre également les fichiers sur les pistes du disque en cas d'incident soudain pendant l'opération de téléchargement. Le module JEEP contient également un gestionnaire de redémarrage dont la responsabilité est de

changer les niveaux d'exécution des fichiers du Juke box une fois que le gestionnaire automatique de fichiers a déterminé qu'une mise à jour du Juke box s'était produite. Le module JEEP.DLL génère également un fichier MISDEPS.DAT, lorsque la détection d'une dépendance manquante a été

5 faite. Ce fichier contient des lignes de la forme NEEDPARSER.DLL → version 2.0 ; NEEDLEXY.DLL → version 2.0, etc, ..... . Ce fichier permet au serveur par lecture de ce fichier MISDEPS.DAT de déterminer les modules sur le juke box et de les recharger.

D'autres modifications à la portée de l'homme de métier font

10 également partie de l'esprit de l'invention.

## **REVENDEICATIONS**

1. Système de téléchargement d'objets ou de fichiers pour mise à  
 5 jour de logiciels caractérisé en ce que l'architecture du système d'exploitation  
 comporte une répartition des différentes tâches en modules logiciels reliés  
 entre eux par des liens dynamiques ou constitués de sous-programmes  
 exécutables présentant des liens de dépendance principale avec d'autres  
 parties du système d'exploitation, chacun des modules étant constitué de  
 10 fichiers d'objets ou de bibliothèques représentées par des bibliothèques de liens  
 dynamiques "dynamic links library" organisées entre elles selon une pluralité  
 de niveaux de dépendance décrits dans leurs attributs respectifs.

2. Système de téléchargement d'objets ou de fichiers pour mise à  
 jour de logiciels selon la revendication 1, caractérisé en ce que les attributs  
 15 d'un objet ou d'une bibliothèque indiquent le numéro de version et les  
 dépendances de l'objet par rapport à d'autres objets.

3. Système de téléchargement d'objets ou de fichiers pour mise à  
 jour de logiciels selon la revendication 2, caractérisé en ce que les attributs  
 indiquent le niveau attribué aux modules.

20 4. Système de téléchargement d'objets ou de fichiers pour mise à  
 jour de logiciels selon une des revendications 1 à 3, caractérisé en ce que  
 les différentes tâches comprennent une tâche principale qui comporte un  
 module (JUKECORE) qui est destiné à charger les bibliothèques de liens  
 dynamiques (DLL), à initialiser le noyau du programme, à initialiser le  
 25 module de gestion des graphiques (GFX), à initialiser le module de  
 chargement des bibliothèques (WDLL), à charger le module Télécom des tâches  
 de télécommunications (TELECOM.DLL) et à lancer le programme  
 d'interpréteur d'écran en tant que tâche principale.

5. Système de téléchargement d'objets ou de fichiers pour mise à  
 30 jour de logiciels selon la revendication 4, caractérisé en ce que le

programme est subdivisé en un certain nombre de modules définissant une tâche spécifique au terminal.

5 6. Système de téléchargement d'objets ou de fichiers pour mise à jour de logiciels selon la revendication 5, caractérisé en ce que cette tâche spécifique est celle correspondant à un juke-box et comporte une tâche principale spécifique, un certain nombre de modules, définissant les fonctions "fenêtres" de l'affichage, qui sont les suivants :

- un module d'exploitation des signaux souris ou de l'écran tactile ;
- un module d'exploitation des messages échangés entre les objets ;
- 10 - un module FIL.DJL de gestion des fichiers sur disques ;
- un module FILIO.DJL de lecture-écriture de fichiers sur disques ;
- un module JSTRUCT.DJL de surveillance de tous les événements générés par les équipements, tels que l'écran tactile, la carte son, l'interface de traitement des équipements monétaire.

15 7. Système de téléchargement d'objets ou de fichiers pour mise à jour de logiciels selon la revendication 5 ou 6, caractérisé en ce que la tâche principale spécifique de l'application juke-box comporte un module (SILOAD.DLL) contenant la librairie du programme chargeur destinée à vérifier les versions de librairies de liens dynamiques requises, de les  
20 charger et d'appeler le module des tâches Télécom pour effectuer le transfert de fichiers.

8. Système de téléchargement d'objets ou de fichiers pour mise à jour de logiciels selon la revendication 7, caractérisé en ce que ce module SILOAD comporte la liste dans un fichier (DLL.DEFAULT) des versions  
25 minimales nécessaires au fonctionnement ainsi que la liste de toutes les fonctions représentées soit par des bibliothèques (DATA JUKEBOX LIBRARY), soit par des fichiers d'objets (DJO, DATA JUKEBOX OBJECT).

9. Système de téléchargement d'objets ou de fichiers pour mise à jour de logiciels selon la revendication 8, caractérisé en ce que chaque objet

ou bibliothèque contient la liste de toutes les fonctions, dont la bibliothèque ou l'objet a besoin, ainsi que les numéros de version et les dépendances.

10. Système de téléchargement d'objets ou de fichiers pour mise à jour de logiciels selon la revendication 4, caractérisé en ce que WDLL fait la  
5 gestion de tous les nouveaux modules, vérifie que les modules téléchargés n'ont aucune dépendance manquante et ont été chargés avec les versions nécessaires.

11. Système de téléchargement d'objets ou de fichiers pour mise à jour de logiciels selon la revendication 7, caractérisé en ce que SILOAD gère  
10 le chargement, des modules spécifiques à la tâche du terminal, à savoir tous les modules DJL déjà énumérés ainsi que les modules de librairie juke-box constitués par WOBJECT qui gère les objets, le mixeur, les achats ; le module WCURSOR qui gère les mouvements du curseur ; le module des DBMAPI qui gère la base de donnée ; le module WFONTs qui gère tous les  
15 types de fonte ; le module PARSER (PARSER Programme d'analyse) qui analyse et génère les écrans à partir du script et vérifie la grammaire à l'aide du module "GRAMMAR.DJL" et le module lexique "LEXY.DJL".

12. Système de téléchargement d'objets ou de fichiers pour mise à jour de logiciels selon la revendication 11, caractérisé en ce que le module  
20 de chargement de librairie SILOAD comporte un module "WINDEF" qui contient la liste des fichiers qu'il faut inclure pour gérer les fenêtres d'un affichage fenêtres prévu sur le moniteur du terminal de type juke-box.

13. Système de téléchargement d'objets ou de fichiers pour mise à jour de logiciels selon la revendication 12, caractérisé en ce que cette liste  
25 d'objets est constituée :

- d'un module objet "OBJET WPSCREEN.DJO" qui permet de définir la page principale sur le moniteur ;

- d'un module "WSCREEN.DJO" qui permet de déterminer dans cette page principale le nombre d'écrans disponibles et ainsi permettre  
30 l'affichage de plusieurs fenêtres ou écrans ;



- d'un module "WIMAGE.DJO" qui permet de déterminer et de définir dans l'écran l'image qu'il utilisera ;

- d'un module "WANIM.DJO" qui permet de définir l'animation lorsque l'image est animée ;

5       - d'un module "WBUTTON.DJO" qui permet de définir et gérer les boutons qui sont utilisés sur l'écran de la page principale ;

- d'un module "WLABEL.DJO" qui permet de créer des étiquettes permettant l'écriture par-dessus un objet ; et

10       - d'un module "WSCROLLER.DJO" qui permet de définir des zones d'affichage à défilement, entre deux points par exemple, vertical, horizontal, diagonal.

14. Système de téléchargement d'objets ou de fichiers pour mise à jour de logiciels selon la revendication 13, caractérisé en ce que l'ensemble de ces modules d'objets gérés par la tâche principale, utilise une librairie  
15       "JHANDLER" qui permet de définir les utilisations fixes des écrans et donc, de déterminer quelles sont les interfaces qui assurent la liaison avec les différents objets définis par les modules précédents.

15. Système de téléchargement d'objets ou de fichiers pour mise à jour de logiciels selon la revendication 7, caractérisé en ce que la tâche  
20       SILOAD lance ou charge le module "XCP" permettant de gérer les tâches de paiement telles que les systèmes de réception de billets ou les unités de paiement à monnaie ou à carte et permet également la sauvegarde des informations de base dans le IBUTTON.

**ABREGE****Inventeur(s) : Guy NATHAN****Dominique DION****Déposant : TECHNICAL MAINTENANCE CORPORATION INC**

5

La présente invention concerne un système de téléchargement d'objets ou de fichiers pour mise à jour de logiciels.

10 Ce système de téléchargement d'objets ou de fichiers pour mise à jour de logiciels est caractérisé en ce que l'architecture du système d'exploitation comporte une répartition des différentes tâches en modules logiciels reliés entre eux par des liens dynamiques ou constitués de sous-programmes exécutables présentant des liens de dépendance principale avec d'autres parties du système d'exploitation, chacun des modules étant  
15 constitué de fichiers d'objets ou de bibliothèques représentées par des bibliothèques de liens dynamiques "dynamic links library" entre elles selon une pluralité de niveaux de dépendance décrits dans leurs attributs respectifs.

Figure 2

SCANNED. # 12

PL 1/2

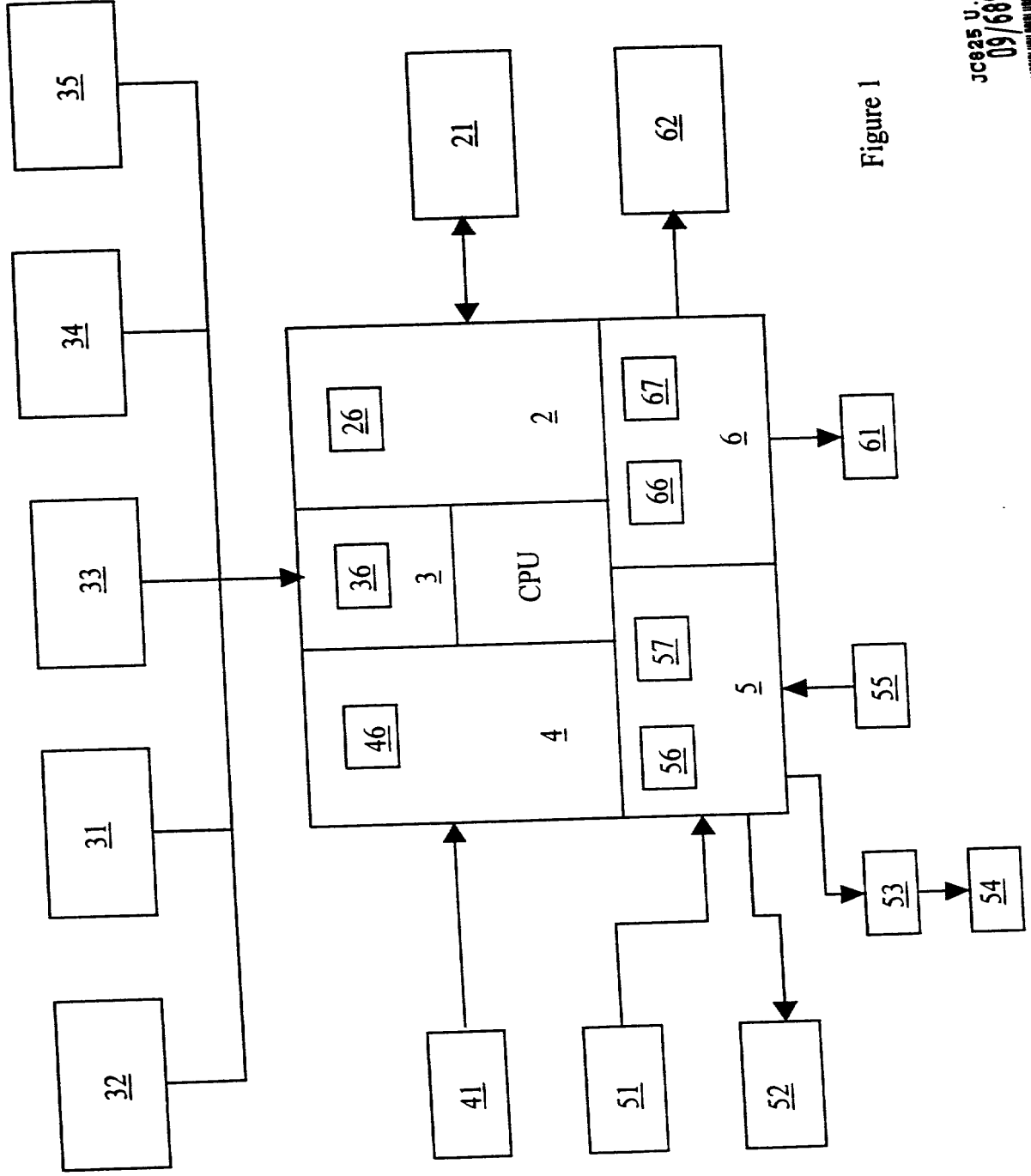


Figure 1

Figure 2

